DO CONVOLUTIONAL NEURAL NETWORKS ACT AS COMPOSITIONAL NEAREST NEIGHBORS?

Chunhui Liu[1,2]*, **Aayush Bansal**[1], **Victor Fragoso**[3], **Deva Ramanan**[1]
[1]Carnegie Mellon University, [2] Peking University, [3] West Virginia University

ABSTRACT

We present a simple approach based on pixel-wise nearest neighbors to understand and interpret the internal operations of state-of-the-art neural networks for pixel-level tasks. Specifically, we aim to understand the synthesis and prediction mechanisms of state-of-the-art convolutional neural networks for pixel-level tasks. To this end, we primarily analyze the synthesis process of generative models and the prediction mechanism of discriminative models. The main hypothesis of this work is that convolutional neural networks for pixel-level tasks learn a fast compositional nearest neighbor synthesis or prediction function. Our experiments on semantic segmentation and image-to-image translation show qualitative and quantitative evidence supporting this hypothesis.

1 INTRODUCTION

Convolutional neural networks (CNNs) have revolutionized computer vision, producing impressive results for discriminative tasks such as image classification and semantic segmentation. More recently, they have also produced startlingly impressive results for image generation through generative models. However, in both cases, such feed-forward networks largely operate as "black boxes." As a community, we are still not able to succinctly state *why* and *how* such feed-forward functions generate a particular output from a given input. If a network fails on a particular input, why? How will a network behave on never-before-seen data? To answer such questions, there is a renewed interest in so-called *explainable AI*[1]. The central goal in this (re)invigorated space is the development of machine learning systems that are designed to be more interpretable and explanatory.

Explanation-by-correspondence: One attractive approach to interpretability stems from case-based reasoning or "explanation-by-example" (Lipton, 2016). Such an approach dates back to classic AI systems that predict medical diagnoses or legal judgments that were justified through case studies or historical precedent (Aamodt & Plaza, 1994). For example, radiologists can justify diagnoses of an imaged tumor as 'malignant' by reference to a previously-seen example (Caruana et al., 1999). However, this approach can generate only N explanations given N training exemplars. Our work demonstrates that deep networks can generate exponentially more explanations through *composition*: e.g., this *part* of the image looks like this *part* of exemplar A, while another *part* looks like that *part* of exemplar B. We term this "explanation-by-correspondence", since our explanations provide detailed correspondence of parts (or even pixels) of a query image to a set of exemplars.

Spatial prediction: In this work, we focus on the class of CNNs designed to make predictions at each image pixel. Many problems in computer vision can be cast in this framework, e.g., semantic segmentation, depth estimation, image synthesis, and image translation. We explore a simple hypothesis for explaining the behavior of such networks: *they operate by cutting-and-pasting image patches found in training data*. Consider the top row of Fig. 1, where we visualize the output of Isola et al. (2016)'s translation network trained to synthesize images of building facades from label masks. Why does the network generate the strange diagonal gray edge at the top? To answer this question, we visualize image pixels extracted from the closest-matching nearest-neighbor (NN) patches found in the training data. Remarkably, NN-synthesis looks quite similar to the CNN output, providing a clear explanation for the synthesized corner artifact.

*Work done as a summer intern at Carnegie Mellon University.

[1]http://www.darpa.mil/program/explainable-artificial-intelligence

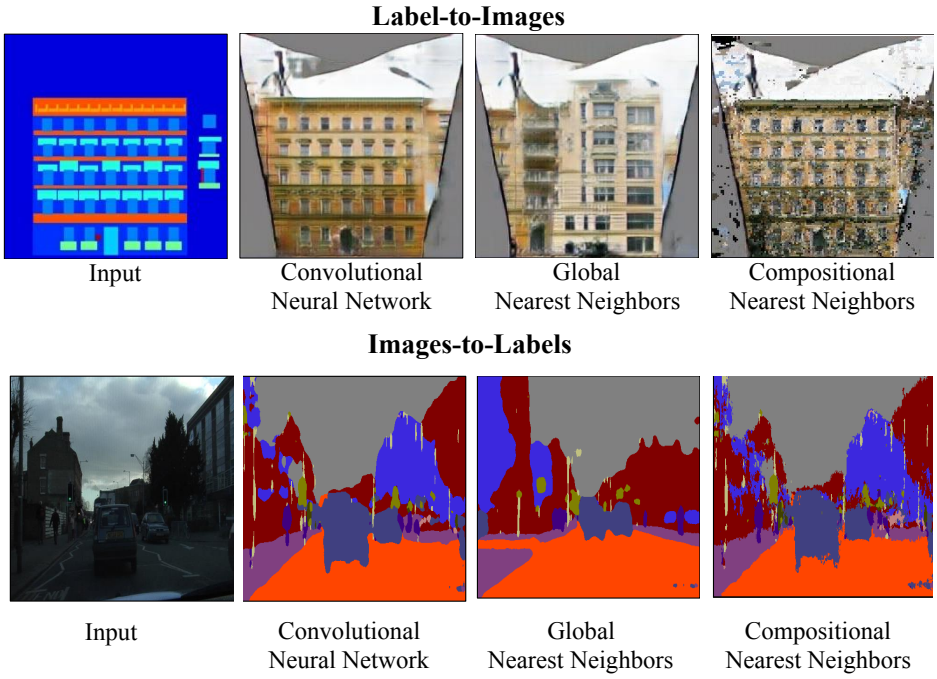

Figure 1: We propose a non-parametric method to *explain and modify* the behavior of convolutional networks, including those that classify pixels and those that generate images. For example, given the label mask on the top, why does a network generate strange gray border artifacts? Given the image on the bottom, how is a network able to segment out the left-most telephone pole in shadow? We advocate an "explanation-by-example" approach to interpretation (Caruana et al., 1999). Next to the CNN output, we show the closest-matching training exemplar image. It appears to provide coarse explanations of such behaviors, though the quality of the output is still lacking (e.g., additional cars are hallucinated in the bottom row). One the right, we show the output obtained through a *compositional nearest-neighbor* operation that simply (1) matches input patches to those in the training set and (2) returns the corresponding output label. This means that the output is created by cutting-and-pasting (composing) patches of training images. To ensure that inconsistent patches are not composed together, one needs to match patches using an embedding that captures both global semantics (e.g., architectural styles) and local structure (e.g., windows versus doors). We demonstrate that local convolutional neighborhoods of feature activations produce such rich embedding. Such a perspective allows one to *explain* errors and *modify* the biases of a network by changing the set of image patches used for non-parametric matching.

Compositional nearest-neighbors: Our central thesis is consistent with recent work on network memorization (Zhang et al., 2016), but notably, naive memorization fails to explain how and why networks generalize to never-before-seen data. We explain the latter through *composition*: the synthesized output in Fig. 1 consists of image patches copied from different training images. Given a database of N training images with K pixels each, global nearest-neighbors can produce N possible output images. On the other hand, compositional nearest-neighbors can produce $(NK)^K$ outputs, an exponentially larger set of outputs. Each output image can be obtained by independently matching each of the K patches in the input query to one of NK patches in the training set. However, many of these outputs may be unrealistic. For example, one should not synthesize a facade by composing a door above a window. To ensure global consistency, one needs to match patches using a carefully-tuned metric that captures such global knowledge. But where do we obtain such a metric?

Patch embeddings: Much past work has demonstrated that intermediate feature activations of a neural network can be interpreted as *global* embeddings for comparing entire images. For example, Sharif Razavian et al. (2014); Devlin et al. (2015) show that the penultimate ("FC7") layer of image classification networks learn embeddings of images that produce remarkably accurate nearest-neighbors. We apply this observation to spatial prediction networks in order to learn local embed-

dings of image *patches* or even pixels. These embeddings are quite rich in that they encode semantic knowledge that is both local (geometric structures centered at the pixel) and global (e.g., color and architectural style of the entire facade).

Correspondence and bias: Beyond being a mechanism for interpretation, we demonstrate that compositional NN matching is a viable algorithm that may approach the accuracy of highly-tuned CNNs. Although slower than a feed-forward net, compositional matching is attractive in two respects: (1) It provides spatial correspondences between pixels in the predicted output and pixels in the training set. Spatial correspondences may be useful in practical applications such as label transfer (Liu et al., 2011). (2) Implicit biases of the network can be explicitly manipulated by *changing* the set of images used for matching – it need not be the same set used for training the network. As an illustrative example, we can force a pre-trained image generation network to predict European or American building facades by restricting the set of images used for matching. Such a manipulation may, for example, be used to modify a biased face recognition network to process genders and races in a more egalitarian fashion.

Contribution: We introduce a **C**ompositional **N**earest **N**eighbors pipeline for interpreting and modifying the behavior of **C**onvolutional **N**eural **N**etworks. Specifically, we demonstrate that CNNs appear to work by memorizing image patches from training data, and then composing them into new configurations. To make compositional matching viable, CNNs learn a local embedding of image patches that captures both global and local semantics. An accurate local embedding is crucial in order to efficiently process an exponentially-large set of potential outputs. We validate our hypothesis on state-of-the-art networks for image translation and semantic image segmentation. Finally, we also show evidence that compositional matching can be used to predict activations of internal layers, generate spatial correspondences, and manipulate the implicitly-learned biases of a network.

2 Related Work

We broadly classify networks for spatial prediction into two categories: (1) discriminative prediction, where one is seeking to infer high-level semantic information from RGB values; and (2) image generation, where the intent is to synthesize a new image from a given input "prior". There is a broad literature for each of these tasks, and here we discuss the ones most relevant to ours.

Discriminative models: An influential formulation for state-of-the-art spatial prediction tasks is that of fully convolutional networks (Long et al., 2015). These have been used for pixel prediction problems such as semantic segmentation (Long et al., 2015; Hariharan et al., 2015; Ronneberger et al., 2015; Bansal et al., 2017a; Chen et al., 2016), depth/surface-normal estimation (Bansal et al., 2016; Eigen & Fergus, 2015), or low-level edge detection (Xie & Tu, 2015; Bansal et al., 2017a). Substantial progress has been made to improve the performance by employing deeper architectures (He et al., 2015), or increasing the capacity of the models (Bansal et al., 2017a), or utilizing skip connections, or intermediate supervision (Xie & Tu, 2015). However, we do not precisely know what these models are actually capturing to do pixel-level prediction. In the race for better performance, the interpretability of these models has been typically ignored. In this work, we focus on interpreting encoder-decoder architectures for spatial classification (Ronneberger et al., 2015; Badrinarayanan et al., 2017).

Image generation: Goodfellow et al. (2014) proposed a two-player min-max formulation where a generator G synthesized an image from random noise z, and a discriminator (D) is used to distinguish the generated images from the real images. While this Generative Adversarial Network (GAN) formulation was originally proposed to synthesize an image from random noise vectors z, this formulation could also be used to synthesize new images from other priors, such as, a low resolution image or label mask by treating z as an explicit input to be conditioned upon. This conditional image synthesis via generative adversarial formulation has been well utilized by multiple follow-up works to synthesize a new image conditioned on a low-resolution image (Denton et al., 2015), class labels (Radford et al., 2015), and other inputs (Isola et al., 2016; Zhu et al., 2017). While the quality of synthesis from different inputs has rapidly improved in recent history, interpretation of GANs has been relatively unexplored. In this work, we examine the influential Pix2Pix network Isola et al. (2016) and demonstrate an intuitive non-parametric representation for explaining its impressive results.

Interpretability: There is a substantial body of work (Zeiler & Fergus, 2014; Mahendran & Vedaldi, 2015; Zhou et al., 2014; Bau et al., 2017) on interpreting general convolutional neural networks (CNNs). The earlier work of Zeiler & Fergus (2014) presented an approach to understand and visualize the functioning of intermediate layers of CNN. Mahendran & Vedaldi (2015) proposed to invert deep features to visualize what is learned by CNNs, similar to inverting HOG features to understand object detection (Vondrick et al., 2013). Zhou et al. (2014) demonstrated that object detectors automatically pop up while learning the representation for scene categories. Krishnan & Ramanan (2016) explored interactive modification of a pre-trained network to learn novel concepts, and recently Bau et al. (2017) proposed to quantify interpretability by measuring scene semantics such as objects, parts, texture, material etc. Despite this, understanding the space of pixel-level CNNs is not well studied. The recent work of PixelNN (Bansal et al., 2017b) focuses on high-quality image synthesis by making use of a two-stage matching process that begins by feed-forward CNN processing and ends with a nonparametric matching of high-frequency detail. We differ in our focus on interpretability rather than image synthesis, our examination of networks for both discriminative classification and image synthesis, and our simpler single-stage matching process that does not require feed-forward processing.

Compositionality: The design of part-based models (Crandall et al., 2005; Felzenszwalb et al., 2008), pictorial structures or spring-like connections (Fischler & Elschlager, 1973; Felzenszwalb & Huttenlocher, 2005), star-constellation models (Weber et al., 2000; Fergus et al., 2003), and the recent works using CNNs share a common theme of *compositionality*. While the earlier works explicitly enforce the idea of composing different parts for object recognition in the algorithmic formulation, there have been suggestions that CNNs also take a compositional approach (Zeiler & Fergus, 2014; Krishnan & Ramanan, 2016; Bau et al., 2017). We see compositional embeddings as rather different than compositional objects/parts. Using (Hinton, 1986)'s terminology, embeddings can be viewed as "distributed representations", while objects/parts can be viewed as "sparse representations". Much past work has argued that distributed representations are central to the success of deep networks (LeCun et al., 2015). We agree and posit that this is one reason why CNNs outperform classic hierarchical models of parts/objects. Specifically, Girshick et al. (2015) point out that classic part models can be implemented as CNNs with sparse activations, where individual neurons correspond to individual part responses. In practice, many neurons are not interpretable when examined individually, as pointed out by Zhou et al. (2014). An embedding perspective offers one solution that does not require individual dimensions to be meaningful - e.g., nearest neighbors in an embedding will not change if one applies a well-behaved linear transformation (e.g., rotation) to the embedding space. This is consistent with past work (Szegedy et al., 2013) that suggests that that linear combinations of activations are equally as informative as the original activations. Finally, if high-level activations represent objects, how can 4K activations (e.g., the typical dimension of FC7) represent 30K+ objects (Biederman, 1987)? Our central thesis is that activations do *not* correspond to individual objects/parts, but rather the dimensions of a local embedding space in which objects/parts are points (matchable with nearest-neighbors).

3 Compositional Nearest Neighbors

We now introduce our method to interpret various fully convolutional networks designed for pixel-level tasks.

Global nearest-neighbors: Our starting point is the observation that classification networks can be interpreted as *linear* classifiers defined on *nonlinear* features extracted from the penultimate layer (e.g., "FC7" features) of the network. We formalize this perspective with the following notation:

$$\text{Label}(x) = k^* \quad \text{where} \quad k^* = \underset{k \in \{1 \ldots K\}}{\operatorname{argmax}} \; w_k \cdot \phi(x), \qquad \text{[K-way classification]} \tag{1}$$

where $\phi(x) \in \mathbb{R}^N$ corresponds to the penultimate FC7 features computed from input image x. Typically, the parameters of the linear classifier $\{w_y\}$ and those of the feature encoder $\phi(\cdot)$ are trained on large-scale supervised datasets:

$$\mathcal{D} = \{(x_n, y_n)\}. \quad \text{[Training database]} \tag{2}$$

Devlin et al. (2015) make the observation that penultimate features $\phi(x)$ can be interpreted as an *embedding* in $\mathbb{R}^N$. By extracting such embeddings for training images x_n, Devlin et al. (2015) build

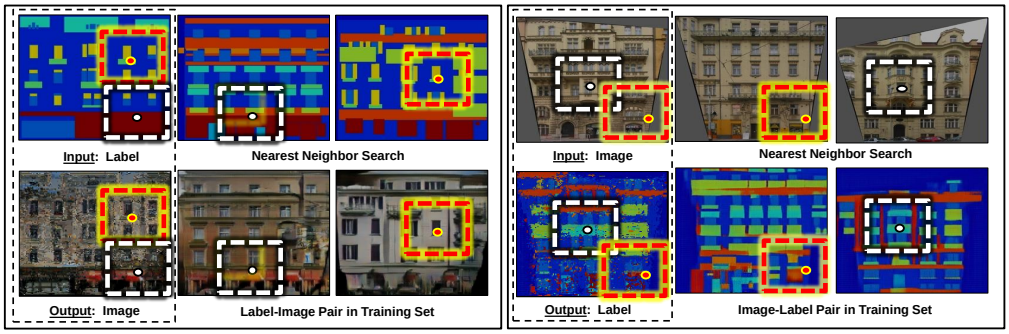

Figure 2: **Overview of pipeline:** Given an input label or image (top-left of each box), our approach extracts an embedding for each pixel. We visualize two pixels with a yellow and white dot. The embedding captures both local and global context, which are crudely visualized with the surrounding rectangular box. We then find the closest matching patches in the training set (with a nearest neighbor search), and then report back the corresponding pixel labels to generate the final output (bottom-left of each box). We visualize an example for label-to-image synthesis on the **left**, and image-to-label prediction on the **right**.

a nonparametric nearest-neighbor (NN) predictor for complex tasks such as image captioning. We write this as follows:

$$\text{Label}(x) = y_{n^*} \quad \text{where} \quad n^* = \underset{n}{\text{argmin}}\, \text{Dist}\Big(\phi(x), \phi(x_n)\Big). \qquad \text{[Global NN]} \qquad (3)$$

Importantly, the above NN classifier performs quite well even when feature encoders $\phi(\cdot)$ are trained for classification rather than as an explicit embedding. In some sense, deep nets seem to implicitly learn embeddings upon which simple linear classifiers (or regressors) operate. We argue that such a NN perspective is useful in interpreting the predicted classification since the *corresponding* training example can be seen as a visual "explanation" of the prediction - e.g., the predicted label for x is "dog" because x looks similar to training image x_{n*}.

Pixel nearest-neighbors: We now extend the above observation to pixel-prediction networks that return back a prediction for each pixel i in an image:

$$\text{Label}_i(x) = k^* \quad \text{where} \quad k^* = \underset{k \in \{1 \ldots K\}}{\text{argmax}}\, w_k \cdot \phi_i(x). \qquad \text{[K-way pixel classification]} \qquad (4)$$

We write $\text{Label}_i(\cdot)$ for the label of the i^{th} pixel and $\phi_i(\cdot)$ for its corresponding feature vector. Because we will also examine pixel-level prediction networks trained to output a continuous value, we write out the following formulation for pixel-level regression:

$$\text{Predict}_i(x) = W\phi_i(x) \quad \text{where} \quad W \in \mathbb{R}^{M \times N}. \qquad \text{[Pixel regression]} \qquad (5)$$

For concreteness, consider a Pix2Pix (Isola et al., 2016) network trained to regress RGB values at each pixel location. These predictions are obtained by convolving features from the penultimate layer with filters of size $4 \times 4 \times 128$. In this case, the three filters that generate R,G, and B values can be written as a matrix W of size $M \times N$, where $M = 3$ and $N = 4 * 4 * 128 = 2048$. Analogously, $\phi_i(x)$ corresponds to N dimensional features extracted by reshaping local 4×4 convolutional neighborhoods of features from the penultimate feature map (of size $H \times W \times 128$). We now can perform nearest-neighbor regression to output pixel values:

$$\text{Predict}_i(x) = y_{n^*,m^*} \quad \text{where} \quad (n^*, m^*) = \underset{n,m}{\text{argmin}}\, \text{Dist}\Big(\phi_i(x), \phi_m(x_n)\Big), \qquad \text{[Comp NN Pixels]}$$

where y_{n^*,m^*} refers to the m^{th} pixel from the n^{th} training image. Importantly, pixel-level nearest neighbors reveals *spatial correspondences* for each output pixel. We demonstrate that these can be used to provide an intuitive explanation of pixel outputs, including an explanation of errors that otherwise seem quite mysterious (see Fig.1 and Fig. 2).

Distance function: We explored different distance functions such as Euclidean and cosine distance. Similar to past work (Devlin et al., 2015), we found that cosine distance consistently performed slightly better, so we use that in all of our experiments.

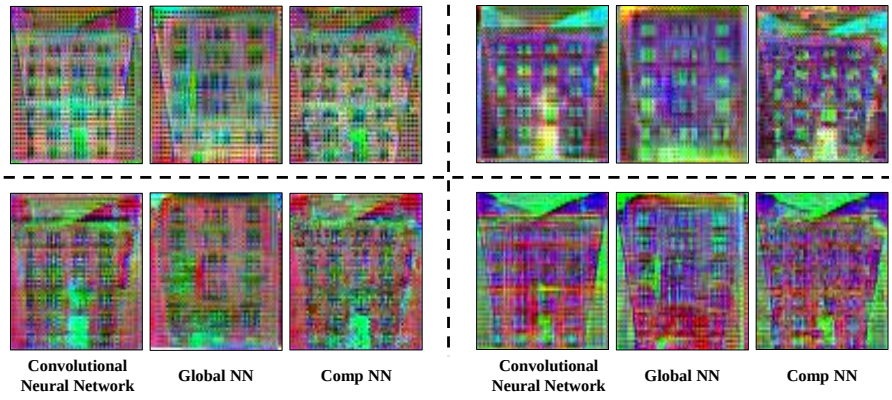

Figure 3: We visualize a non-parametric approach to computing activations from internal layers. By matching to a training database of $decoder_4$ features from Pix2Pix, we can compute activations for the next layer ($decoder_3$) with nearest-neighbors. Each image represents a feature map of 3 continuous channels visualized in the R,G, and B planes. The collective set of 4 images displays 12 out of the 256 channels in $decoder_3$. Global nearest-neighbors (i.e., matching to the training image with the most similar $decoder_4$ layer and returning its associated $decoder_3$ layer) produces poor matches, but compositional-pasting matches together from different exemplars produce activations that are nearly identical to those computed by the underlying CNN.

Convolutional embeddings: We now extend our compositional nearest-neighbor formulation to internal convolutional layers. Recall that activations a at a spatial position i and layer j can be computed using thresholded linear functions of features (activations) from the previous layer:

$$a_{ij}(x) = \max\Big(0, W\phi_{ij}(x)\Big), \qquad \text{where} \quad a_{ij} \in \mathbb{R}^M,\ \phi_{ij} \in \mathbb{R}^N,\ W \in \mathbb{R}^{M \times N} \tag{6}$$

where we write a_{ij} for the vector of activations corresponding to the i^{th} pixel position from layer j, possibly computed with bilinear interpolation (Long et al., 2015). We write ϕ_{ij} for the local convolutional neighborhood of features (activations) from the previous layer that are linearly combined with bank of linear filters W to produce a_{ij}. For concreteness, let the previous layer be $decoder_4$ from Pix2Pix, and the current layer be $decoder_3$. We can then write, $a_{ij} \in R^M$ where $M = 256$ and $\phi_{ij} \in R^N$ where $N = 4 * 4 * 512 = 8192$. We similarly posit that one can produce approximate activations by nearest neighbors. Specifically, let us run Pix2Pix on the set of training images, and construct a dataset of training patches with features $\phi_{ij}(x_n)$ as data and corresponding *activation vectors* $a_{ij}(x_n)$ as labels. We can then predict activation maps for a query image x with NN:

$$a_{ij}(x) = a_{m^*,j}(x_{n^*}) \quad \text{where} \quad (m^*, n^*) = \underset{m,n}{\operatorname{argmin}} \operatorname{Dist}\Big(\phi_{ij}(x), \phi_{mj}(x_n)\Big). \quad \text{[Predicting Activations]}$$

Notably, this is done without requiring explicit access to the filters W. Rather such responses are implicitly encoded in the training dataset of patches and activation labels. We show that such an approach actually produces reasonable activations (Fig. 3).

Patch nearest-neighbors: The previous paragraph demonstrated that composing image patches using embeddings from interior layers could "explain" the behavior of the subsequent layer. We now ask a more ambitious question - could such a procedure "explain" the behavior of *all* subsequent layers? That is, could it produce output predictions that mimic the behavior of the entire network? To do so, we regress the final-layer pixel value from each stored patch:

$$\text{Predict}_{ij}(x) = y_{n^*,m^*} \quad \text{where} \quad (n^*, m^*) = \underset{n,m}{\operatorname{argmin}} \operatorname{Dist}\Big(\phi_{ij}(x), \phi_{mj}(x_n)\Big). \quad \text{[Comp NN Patches]} \tag{7}$$

As written, the above procedure is inefficient because features are interpolated (to pixel resolution) before they are matched to the patch database. Instead, it is natural to match features at their native resolution, and then interpolate the matches. This results in significant speed ups. For example, the

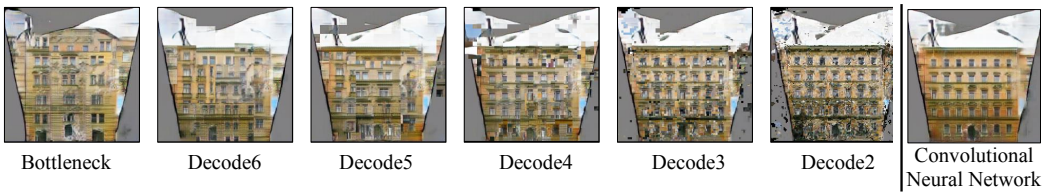

Figure 4: **Adding composition by matching to later layers:** We apply compositional nearest-neighbor matching to features extracted from different layers, starting with the bottleneck layer and progressing to the penultimate deconv2 layer. We match local neighborhoods of convolutional embeddings, which naturally allows for more composition as we use later laters.

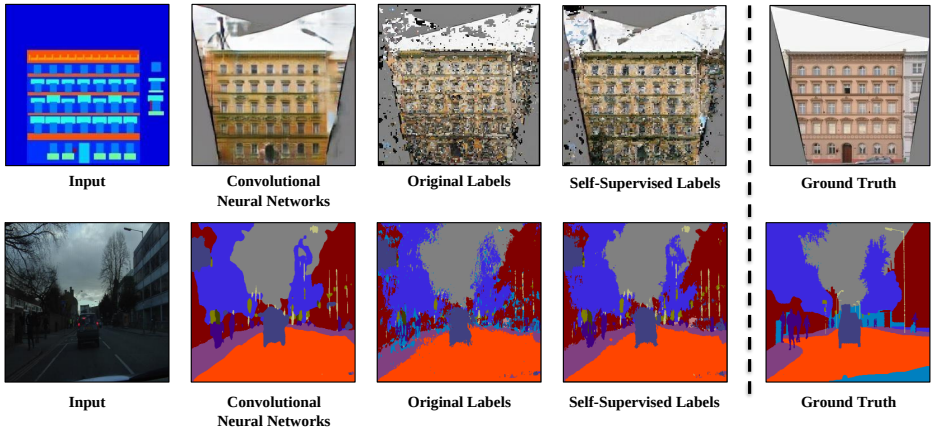

Figure 5: **Original labels v.s. self-supervised labels:** Given the label input on the **left**, we show results of Pix2Pix in the Convolutional Neural Networks column, and non-parametric matching to the training set using the original labels and the predicted "self-supervised" labels of the Pix2Pix network. Generating images with the predicted labels looks smoother, though the qualitative behavior of the network is still explained by the original training labels. We quantify this in our experimental results, and include additional qualitative visualizations of the original and self-supervised labels in Figs. 13 and 14.

bottleneck layer has an activation map of size $1 \times 1 \times 512$. Matching to bottleneck features in the training set is quite fast because it acts as a compact global descriptor for matching entire images. The downside is that the matches are not compositional. By matching to convolutional embeddings extracted from later layers, one can compute progressively more compositional matches, that are initially global, then patch-based, and finally pixel-based (see Fig. 4). In our experiments, we found that such patch embeddings could be used to prune the NN pixel search, significantly speeding up run-time performance (e.g., we first prune the training database to a shortlist of images with similar bottleneck features, and then search through these images for similar patches, and then search through those patches for similar pixels).

Bias modification: Finally, our results suggest that the matching database from Eq.(2) serves as an explicit "associative memory" of a network (Carpenter, 1989). We can explicitly modify the memory by changing the dataset of training images $\{x_n\}$, labels $\{y_n\}$, or both. We experiment with various modifications in our experimental results. One modification that consistently produced smoother visual results was to refine the training labels to those predicted by a network:

$$\{(x_n, y_n)\} \Rightarrow \{(x_n, CNN(x_n))\}. \quad \text{[Self-Supervised Labels]} \tag{8}$$

Such as procedure for "self-supervised" learning is sometimes used when training labels are known to be noisy. From an associative network perspective, we posit that such labels capture a more faithful representation of a network's internal memory. Unless otherwise specified, all results make use of the above matching database. We visualize the impact of self-supervised labels in Fig. 5.

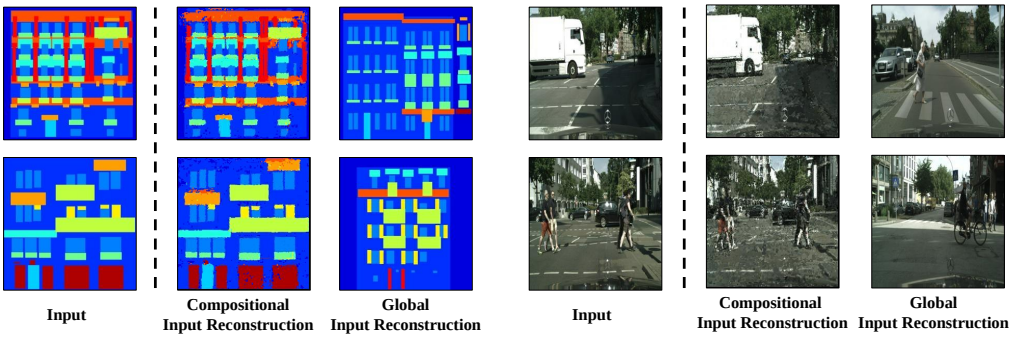

Figure 6: **Reconstruction:** We can use our nonparametric matching framework to generate reconstructions by replacing the exemplar target label y_n with the exemplar input image x_n. This can be done for both image generation and discrete label prediction. We find that, perhaps surprisingly, pixel embeddings contain enough local information to reconstruct the input pixel. We show additional results in Fig. 10.

Reconstruction: We found that replacing the label y_n with the input image x_n is a helpful diagnostic for visualization. This illustrates the ability of the learned embedding and compositional matching framework to reconstruct the input query. The reconstructed input for a global NN match is simply the best-matching exemplar *input* image (see Fig. 6). We find that, perhaps surprisingly, pixel embeddings contain enough local information to reconstruct the input pixel. We show additional results in our experimental results.

$$\{(x_n, y_n)\} \Rightarrow \{(x_n, x_n)\} \quad \text{[Reconstruction]} \tag{9}$$

3.1 Sufficient Statistics for Pixel-level Tasks

We now discuss information-theoretic properties of the introduced nearest-neighbor embedding $\phi_i(\cdot)$ presented in Sec. 3. Specifically, we show that this embedding produces sufficient statistics (Tishby & Zaslavsky, 2015; Shwartz-Ziv & Tishby, 2017; Achille & Soatto, 2017) for various pixel level tasks.

Global embeddings: We begin with the simpler case of predicting a global class label y from an image x. If we assume that the input image x, global feature embedding $\phi(x)$, and output label y form a Markov chain $x \to \phi(x) \to y$, we can write the following:

$$p(y \mid \phi(x),\ x) = p(y \mid \phi(x)). \qquad \text{[Sufficiency]} \tag{10}$$

As Achille & Soatto (2017) show, standard loss functions in deep learning (such as cross-entropy) search for an embedding that minimizes the entropy of the label y given the representation $\phi(x)$. This observation explicitly shows that the embedding $\phi(x)$ is trained to serve as a *sufficient representation* of the data x that is rich enough in information to predict the label y. In particular, if the learned embedding satisfies the above Markov assumption, the prediction y will not improve even when given access to the raw data x.

Pixel-wise embeddings: Spatial networks predict a set of pixel-wise labels $\{y_i\}$ given an input image. If we assume that the pixel-wise labels are conditionally independent given $\{\phi_i(x)\}$ and x, then we can write the joint posterior distribution over labels with the following product:

$$p(\{y_i\} \mid \phi(x),\ x) = \prod_i p(y_i \mid \phi(x),\ x) \qquad \text{[Conditional Spatial Independence]} \tag{11}$$

$$= \prod_i p(y_i \mid \phi_i(x)), \qquad \text{[Sufficiency]} \tag{12}$$

where $\phi_i(x)$ are the sufficient statistics needed to predict label y_i, and $\phi(x) = \{\phi_i(x)\}$ is the aggregate set of these sufficient statistics. One can similarly show that pixel-wise cross-entropy

losses jointly minimize the entropy of the labels y_i given $\phi_i(x)$. This suggests that pixel-wise features $\phi_i(x)$ do serve as a remarkably rich characterization of the image. This characterization includes both global properties (e.g., the color of a building facade being synthesized) as well as local properties (e.g., the presence of a particular window ledge being synthesized). Importantly, this requires the conditional independence assumption from Eq. (11) to be conditioned on the entire image x rather than just the local pixel value x_i (from which it would be hard to extract global properties).

An interesting observation from the factorization shown in Eq. (12) is that we can synthesize an image by predicting pixel values independently. Thus, this theoretical observation suggests that a simple nearest-neighbor regression for every output pixel can synthesize plausible images.

Multi-layered representations: The above pixel-wise formulation also suggests that internal feature layers serve as sufficient representations to predict activations for subsequent layers. Interestingly, skip connections that directly connect lower layers to higher layers break the Markov independence assumption. In other words, skip connections suggest that higher layer features often do *not* serve as sufficient representations, in that subsequent predictions improve when given access to earlier layers. However, Eq.(12) technically still holds so long as we write $\phi_i(x)$ for the *concatenated* representation including lower-level features. For example, in Pix2Pix, we write $\phi_i(x) \in \mathbb{R}^N$ where $N = 4 * 4 * (64 + 64) = 2048$, where the second set of 64 channel features are copied from the first *encoder* layer. In the next Section, we show qualitative and quantitative experiments supporting this analysis.

Prior work on embeddings: Now that our framework has been described both algorithmically and theoretically, we compare it to a large body of related work. The idea that intermediate CNN layers learn embeddings is not new. This dates back at least to Caruana et al. (1999), and was popularized in recent history with (Sharif Razavian et al., 2014; Devlin et al., 2015). Indeed, much contemporary work makes use of "off-the-shelf" CNN layers as features, where earlier layers tend to encode more generic feature representations (Zeiler & Fergus, 2013). However, such representations are typically *global* and refer to the entire image. Alternatively, one can extract *local* pixel-based feature representations, but these are typically defined by $1x1$ slices of a convolutional feature map (Hariharan et al., 2015; Long et al., 2014). Our theoretical analysis, while quite straightforward, shows that the optimal local representation (in terms of sufficiency) is given by a *convolutional* neighborhood of overlapping activations. Finally, we show that *compositional* matching with such local embeddings significantly outperforms global matching (see Fig. 3), and rivals the accuracy of feedforward CNN predictions (Table 1).

4 EXPERIMENTS

We now present experimental results for discriminative networks trained for semantic segmentation, as well as generative networks trained for image synthesis. The goal of the experiments is to show that images generated with a simple NN regression are a good way to interpret the internal operations of a convolutional neural network.

Networks: We use SegNet (Badrinarayanan et al., 2017), a recent state-of-the-art network for image segmentation, and Pix2Pix (Isola et al., 2016), a state-of-the-art network for conditional image synthesis and translation. We evaluate our findings for multiple datasets and tasks on which the original networks were trained. These include tasks such as synthesizing facades from architectural labels and vice versa, predicting segmentation class labels from urban RGB images, and synthesizing Google maps from aerial or satellite views.

Semantic segmentation: We use the CityScape (Cordts et al., 2016) and CamVid (Brostow et al., 2008; 2009) datasets for the task of semantic segmentation. Both datasets are annotated with semantic class labels for outdoor images collected from a car driving on the road. We use SegNet (Badrinarayanan et al., 2017) for CamVid sequences, and Pix2Pix (Isola et al., 2016) for the CityScape dataset. Figure 7 shows qualitatively results for these datasets. We can observe in Figure 7 that the compositional NN produces a nearly-identical result (CompNN column) to the one produced by the network (Pix2Pix and SegNet columns). This suggests that our method enables a good interpretation of discriminative deep networks on pixel-level classification.

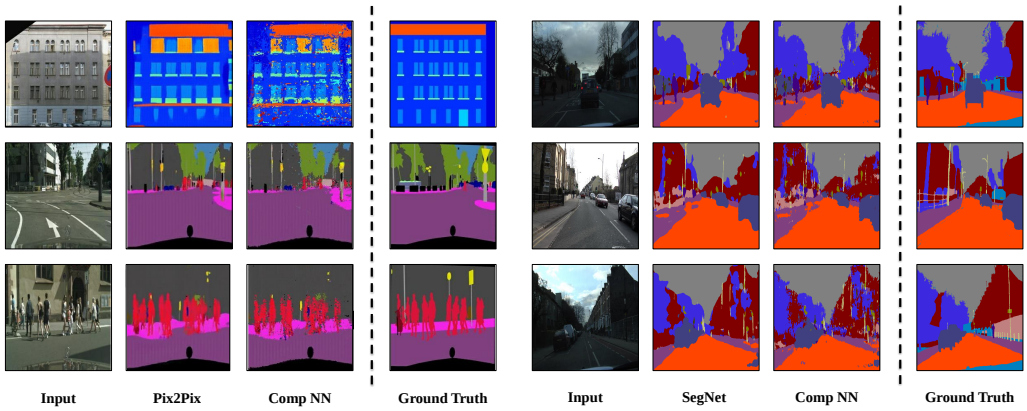

Figure 7: **Semantic segmentation:** The results of semantic segmentation on cityscape and CamVid dataset. This result suggests the following observations. First, the difference between images generated by generative networks (Pix2Pix and SegNet columns) and NN embedding (CompNN column) is surprisingly small. Thus, our method can perfectly interpret discriminative deep networks on pixel-level classification. Second, we can notice some noise edges with a high gradient (see columns 5-8). This phenomenon can also be used to understand the difficulty of image segmentation task: borders with high gradients are usually hard to classify due to the ambiguous patches in training set.

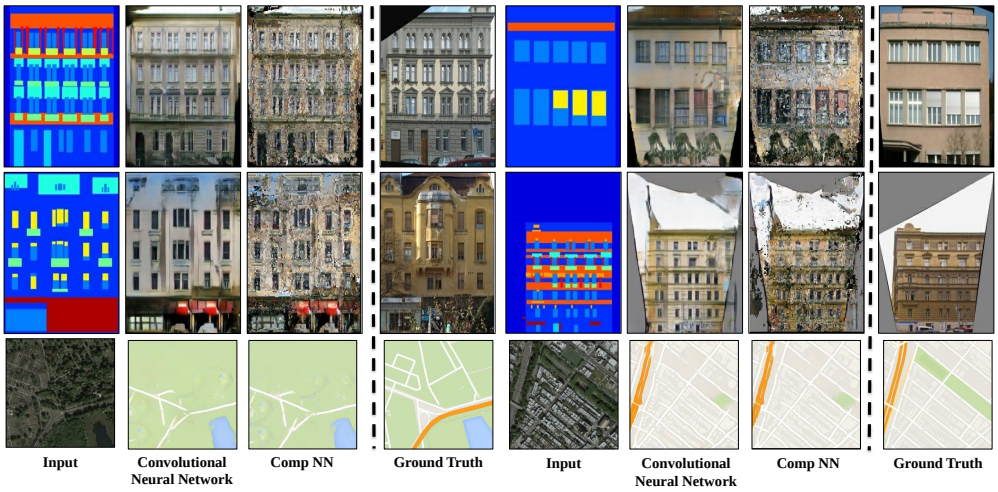

Figure 8: **Image synthesis:** The results suggest that Comp NN (our approach) can also explain the results from Pix2Pix. We conclude this because our approach reproduces color and the structure of the Pix2Pix output, including a few artifacts (e.g., the image cuts in the 6th and 7th columns).

Architectural labels-to-facades: We followed Isola et al. (2016) for this setting and used the annotations from (Tyleček & Šára, 2013). There are 400 training images in this dataset, and 100 images in the validation set. We use the same dataset to generate architectural labels from images of facades. Pix2Pix (Isola et al., 2016) models are trained using 400 images from the training set for both labels-to-facades and vice versa. Figure 8 shows qualitative examples of synthesizing real-world images using pixel-wise nearest neighbor embedding in its first and second rows. We observe that the NN-embedding perfectly explains the generation of deformed edges (see CompNN column), and how the generative architecture is eventually memorizing patches from the training set.

Satellite-to-maps: This dataset contains 1096 training images and 1000 testing images scraped from Google Maps. We use the same settings used by Isola et al. (2016) for this task. Figure 8 qualitatively shows in its third row how Google maps are synthesized from satellite images. We

Figure 9: **Bias modification:** Given the same label input, we show different results obtained by matching to different databases (using an embedding learned by Pix2Pix). By modifying the database to include specific buildings from specific locations, one can introduce and remove implicit biases in the original network (e.g.,one can generate "European" facades versus "American" facades).

can observe that our synthesis (CompNN column) is nearly identical to the image generated by the network (Pix2Pix columns).

Bias modification: Figure 9 shows that the output of nonparametric matching (3rd - 6th columns from left to right) can be controlled through explicit modification of the matching database. In simple words, this experiment shows that we can control properties of the synthesized image by simply specifying the exemplars in the database. We can observe in Figure 9 that the synthesized images preserve the structure (e.g., windows, doors, and roof), since it is the conditional input, but the textural components (e.g., color) change given different databases.

Reconstruction: This experiment shows that the learned embeddings of a CNN also enables the reconstruction of the input images in a compositional fashion (as discussed in Sec. 3). The results of this experiment are shown in Fig. 10. The Figure has the following organization. In the middle columns (enclosed in a box), the Figure shows the input and output images. The first two columns (from left-to-right) show the reconstructions of the *input* images using a global nearest-neighbors and the proposed compositional nearest-neighbors approach. The last two columns show the reconstruction of the *output* images, also using a global nearest-neighbors and the proposed compositional nearest-neighbors approach. We can observe that the reconstructions of the input images using the global nearest-neighbors approach overall resembles the structure of the scene. However, the reconstructions of the input images using the proposed compositional nearest-neighbors reproduce the input scene with a remarkable accuracy. These results suggest that the learned embedding is rich in global and local information to either reconstruct both the input and output images. We can conclude then that CNNs understand an input image by finding the patches from the training images that enable the composition of an image reproducing the input. To the best of our knowledge, this is the first approach that reconstructs an input image using training instances using a learned pixel-embedding.

Correspondence map: Figure 11 shows a correspondence map that explicitly illustrates how an output image is synthesized by cutting-and-pasting patches from training images. We can observe that patches are selected from many different styles of facades, but in such a manner that ensures that the composed output is globally consistent while maintaining the appropriate local structures (such as windows, doors, awnings, etc.). This implies the learned patch embedding captures both global semantics and local structure.

Quantitative evaluation: We present the quantitative analysis of our pixel-wise nearest neighbor approach with an end-to-end pipeline in Table 1. We report classification accuracy of ground truth labels and mean intersection-over-union (IoU) compared to the predicted labels for the task of semantic segmentation. We can observe in Table 1 that compositional matching approaches the accuracy of the baseline CNN, and dramatically outperforms global matching (sometimes by a factor of 2X). Finally, self-supervised labels (SS) overall perform similarly to the original labels (O), but almost consistently help for compositional matching and consistently hurt for global matching. We posit that this is due to the fact that self-supervised labels tend to be overly-smoothed, and so act as a form of spatial regularization that helps compositional matching.

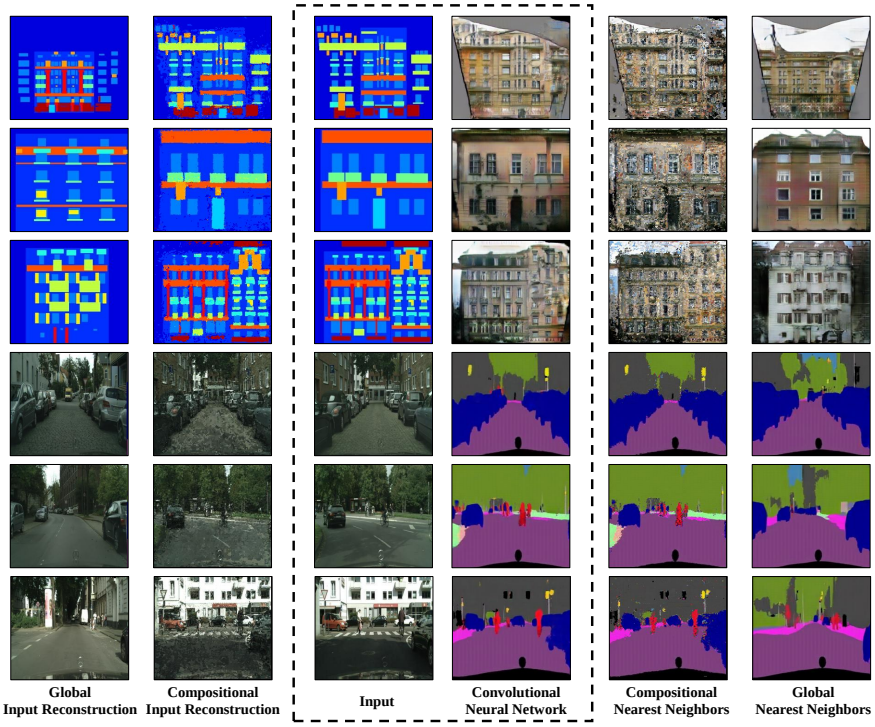

Figure 10: **Reconstruction:** Given the correspondences of NN features from penultimate the layer, we reconstruct the test input images (third column from left-to-right) by using the compositional nearest-neighbor approach: copying and pasting corresponding image patches from the input images of the training set. The reconstructions using a compositional nearest-neighbor approach is shown in the second column, while the reconstructions using a global nearest-neighbor approach is shown in the first column. The learned embedding thus enables not only the reconstruction of the input image, but also of the output image (see the last two columns). These results suggest that the embedding possess not only the information relevant to a specific task, but also semantic information from the original image. We can conclude then that CNNs understand an input image by finding the patches from the training images that enable the composition of an image reproducing the input.

Implementation Details: We used U-net as the generator for Pix2Pix, and used publicly available Tensorflow code for Pix2Pix[2] and SegNet[3]. For a slightly faster computation, we used the *Eigen* Library to implement the cosine distance. For Cityscape dataset, we shortlist 100 global neighborhoods using global bottleneck features for compositional NN searching. This leads to a 30 times speedup. For CamVid dataset, we shortlist 10 global neighborhoods. We can observe in previous results that the quality of generated images is hardly affected. We used 40-threads for these experiments. The average compute time per image is 22 minutes and 13 minutes for Cityscape and CamVid dataset respectively.

5 Discussion

In this paper, we have presented a simple approach based on pixel-wise nearest neighbors to understand and interpret the functioning of convolutional neural networks for spatial prediction tasks. Our analysis suggests that CNNs behave as compositional nearest neighbor operators over a training set of patch-label pairs that act as an associative memory. But beyond simply memorizing, CNNs can generalize to novel data by composing together local patches from different training instances. Also, we argued that networks for pixel-level tasks learn sufficient statistics that enable the gener-

[2] https://github.com/affinelayer/pix2pix-tensorflow
[3] https://github.com/tkuanlun350/Tensorflow-SegNet

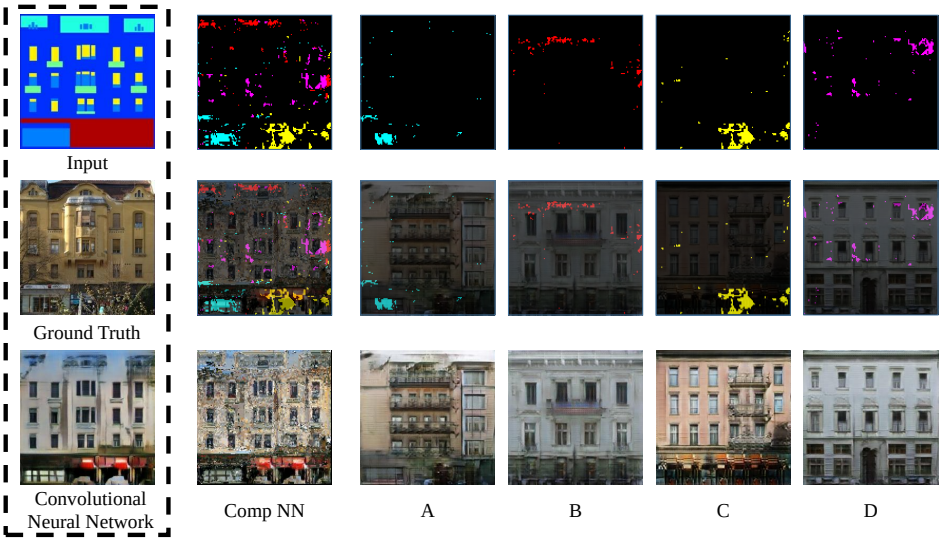

Figure 11: **Correspondence map:** Given the input label mask on the top left, we show the ground-truth image and the output of Pix2Pix below. Why does Pix2Pix synthesize the peculiar red awning from the input mask? To provide an explanation, we use CompNN to synthesize an image by explicitly cutting-and-pasting (composing) patches from training images. We color code pixels in the training images to denote correspondences. For example, CompNN copies doors from training image A (blue) and the red awning from training image C (yellow).

Approach	**Facades**	**CityScape**	**CamVid**	**Facades**	**CityScape**	**CamVid**
	(Mean Pixel Accuracy)			*(Mean IoU)*		
Baseline CNN	0.545	0.735	0.790	0.157	0.217	0.444
Comp NN (SS)	0.505	0.738	0.767	0.137	0.210	0.378
Comp NN (O)	0.493	0.722	0.754	0.134	0.217	0.372
Global Bottleneck NN (SS)	0.324	0.579	0.564	0.057	0.109	0.246
Global Bottleneck NN (O)	0.381	0.585	0.570	0.065	0.133	0.226
Global Decode2 NN (SS)	0.387	0.590	0.659	0.087	0.110	0.287
Global Decode2 NN (O)	0.393	0.600	0.664	0.090	0.136	0.308

Table 1: We compare compositional nearest neighbors (CompNN) to the baseline CNN and different global nearest neighbor approaches, obtained by matching feature maps from different layers (Global-Bottleneck and Global-Decode2). We report mean pixel accuracy and intersection-over-union, where predicted segmentation labels are compared to ground-truth labels. We specifically use the embedding learned by Isola et al. (2016) for Facades-to-Labels (Facades) and CityScape, and embedding learned by Badrinarayanan et al. (2017) for CamVid. On average, CompNN performs 5% worse than the baseline CNN, though in some cases (CityScapes) it performs equally. However, compositional matching dramatically outperforms global matching, sometimes by a factor of 2X (Facade and CityScape IoU). In terms of global matching, the last feature layer (Decode2) strictly outperforms the intermediate Bottleneck layer, but is significantly larger (128^3 versus 512 dimensions). Finally, self-supervised labels (SS) overall perform similarly to the original labels (O), but almost consistently help for compositional matching and consistently hurt for global matching. We posit that this is due to the fact that self-supervised labels tend to be overly-smoothed, and so act as a form of spatial regularization for compositional matching.

ation of pixel predictions. Our analysis and experiments not only support this argument, but also enables example-based explanations of network behavior and explicit modulation of the implicit biases learned by the network. We hope that our framework enables further analysis of convolutional networks from a non-parametric perspective.

REFERENCES

Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7(1):39–59, 1994.

Alessandro Achille and Stefano Soatto. On the emergence of invariance and disentangling in deep representations. *CoRR*, abs/1706.01350, 2017. URL `http://arxiv.org/abs/1706.01350`.

Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for scene segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.

Aayush Bansal, Bryan Russell, and Abhinav Gupta. Marr Revisited: 2D-3D model alignment via surface normal prediction. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Aayush Bansal, Xinlei Chen, Bryan Russell, Abhinav Gupta, and Deva Ramanan. PixelNet: Representation of the pixels, by the pixels, and for the pixels. *arXiv:1702.06506*, 2017a.

Aayush Bansal, Yaser Sheikh, and Deva Ramanan. PixelNN: Example-based image synthesis. *CoRR*, abs/1708.05349, 2017b.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. *CoRR*, abs/1704.05796, 2017.

I. Biederman. Recognition by components: a theory of human image interpretation. *Pyschological review*, 94: 115–147, 1987.

Gabriel J. Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *Proc. of the European Conference on Computer Vision (ECCV)*, pp. 44–57, 2008.

Gabriel J Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009.

Gail A Carpenter. Neural network models for pattern recognition and associative memory. *Neural networks*, 2 (4):243–257, 1989.

Rich Caruana, Hooshang Kangarloo, JD Dionisio, Usha Sinha, and David Johnson. Case-based explanation of non-case-based learning methods. In *Proceedings of the AMIA Symposium*, pp. 212. American Medical Informatics Association, 1999.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR*, abs/1606.00915, 2016.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

David Crandall, Pedro Felzenszwalb, and Daniel Huttenlocher. Spatial priors for part-based recognition using statistical models. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

Emily L. Denton, Soumith Chintala, Arthur Szlam, and Robert Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. *CoRR*, abs/1506.05751, 2015.

Jacob Devlin, Saurabh Gupta, Ross B. Girshick, Margaret Mitchell, and C. Lawrence Zitnick. Exploring nearest neighbor approaches for image captioning. *CoRR*, abs/1505.04467, 2015.

David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 2005.

R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Trans. Comput.*, 22(1), January 1973.

Ross Girshick, Forrest Iandola, Trevor Darrell, and Jitendra Malik. Deformable part models are convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 437–446, 2015.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial networks. *CoRR*, abs/1406.2661, 2014.

Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

Geoffrey E. Hinton. Learning distributed representations of concepts. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, pp. 1–12. Hillsdale, NJ: Erlbaum, 1986.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arxiv*, 2016.

Vivek Krishnan and Deva Ramanan. Tinkering under the hood: Interactive zero-shot learning with net surgery. *CoRR*, abs/1612.04901, 2016.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

Zachary C Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

Ce Liu, Jenny Yuen, and Antonio Torralba. Nonparametric scene parsing via label transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2368–2382, 2011.

Jonathan Long, Ning Zhang, and Trevor Darrell. Do convnets learn correspondence? In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional models for semantic segmentation. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 806–813, 2014.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. *CoRR*, abs/1503.02406, 2015. URL `http://arxiv.org/abs/1503.02406`.

Radim Tyleček and Radim Šára. Spatial pattern templates for recognition of objects with regular structure. In *Proc. German Conference on Pattern Recognition (GCPR)*, Saarbrucken, Germany, 2013.

C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba. HOGgles: Visualizing Object Detection Features. *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2013.

Markus Weber, Max Welling, and Pietro Perona. Towards automatic discovery of object categories. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000.

Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.

Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Proc. of the European Conference on Computer Vision (ECCV)*, pp. 818–833, 2014.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2014.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017.

A Appendix: Additional Experimental Results

A.1 Global Nearest Neighbors

We also present synthesized images using a global nearest neighbors (NN) approach. In this case, we use the global information from the bottleneck features and FC7 features. These bottleneck features can reveal which patches are learned and which training instances have more influence than others.

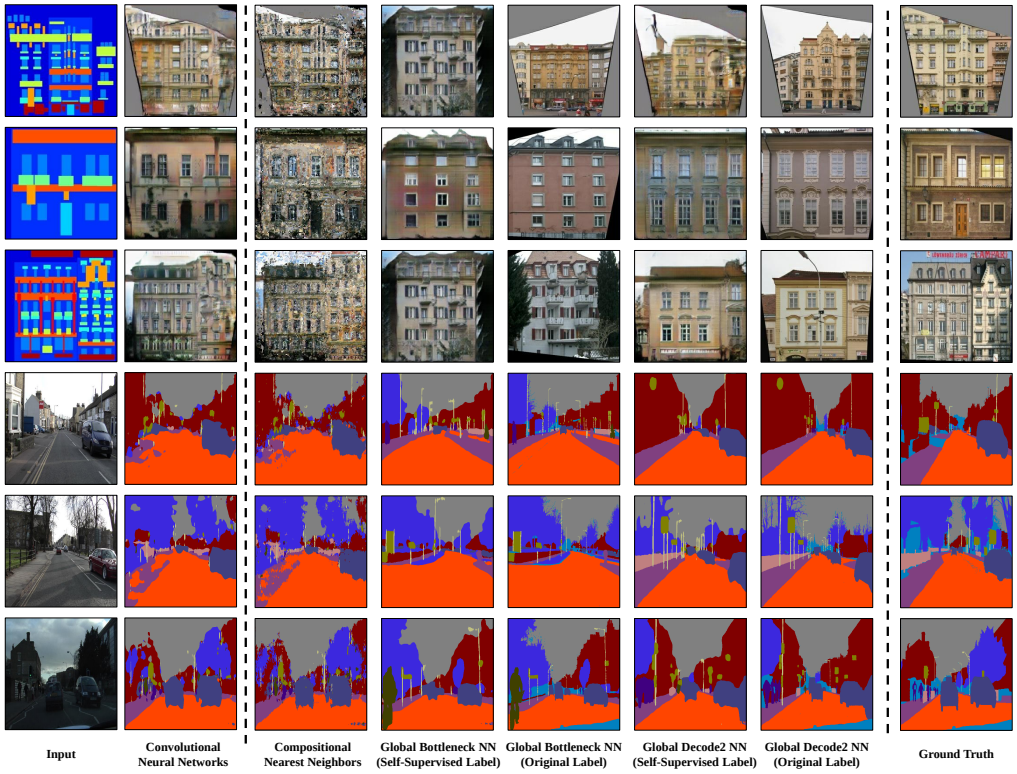

Figure 12: Global NN v.s. Comp NN. We show synthesized images using our CompNN methods and four global NN approaches (global nearest neighbor on bottleneck feature embedding and Decode2 feature embedding using self-supervised labels and original labels respectively). We can observe that (1) compositional nearest neighbor outperforms other global nearest neighbor approaches, (2) using Decode2 features (the penultimate layer) sometimes can generate more similar structures (See row 1,4).

Fig. 12 shows the synthesized images using several global NN approaches and a CompNN approach. We can observe that the results of global NN approaches overall resembles global properties of the output of the Convolutional Neural Network (CNN) and of the CompNN approach. For instance, in the top two rows, the output of the global NN resembles the color of the facade and structural properties of the buildings. Also, in the bottom two rows, we can observe that the global NN overall captures the organization of the scene because many labels in the global NN overlap considerably with the output of the CNN and the ground truth.

A.2 Compositional Nearest Neighbors

In this section, we show more results of our proposed Compositional Nearest Neighbors. Both self-supervised labels and original labels are evaluated in this section. We can observe in Fig.13 that the results of the Compositional Nearest Neighbors (CompNN) approach are quite similar to those of the Convolutional Neural Network. We can also observe that the CompNN produces smoother results when it uses self-supervised labels than when it uses the original ones. Moreover, the self-supervised CompNN method produces results that are more alike to those of the Convolutional Neural Network.

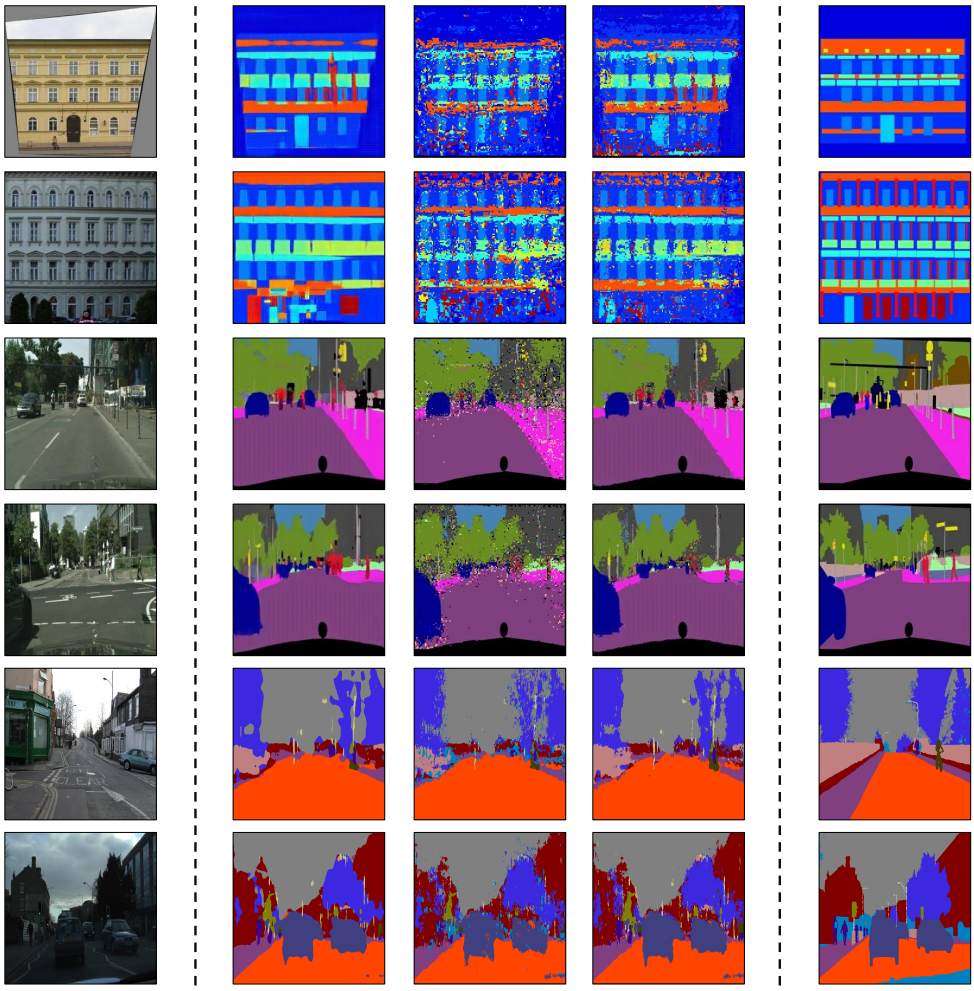

Figure 13: Compositional Nearest Neighbors (CompNN) segmentation results using self-supervised and original labels. Overall, CompNN produces similar results compared with those of the Convolutional Neural Network. In particular, CompNN produces smoother results when it uses self-supervised labels than when it uses the original labels.

A.3 Additional Image Synthesis Results

Fig. 14 shows additional image syntheses using a CNN and CompNN with original and self-supervised labels. As discussed earlier, the CompNN with self-supervised labels produces a smoother image than when it uses the original labels.

B Appendix: Experimental Details

B.1 Computational Complexity

Although Compositional Nearest Neighbors provide insights into the internal operations of a CNN, its computational complexity is very high. In this section, we show some experimental details to speed up the CompNN process. Assume a dataset with N images, each with $H \times W$ pixels, and M filters from the last layer of a CNN. Then, the computational complexity for synthesizing one image using CompNN is

$$O(NH^2W^2M^2). \tag{13}$$

Figure 14: Synthesized images for pixel-wise prediction tasks with a Convolutional Neural Network, and Compositional Nearest Neighbors using self-supervised and original labels.

We now introduce several approaches to speed up searching process. (1) Although Numpy from Python calculates the distance between two features quickly, the iterations for synthesizing a pixel are slow. To alleviate this, we implemented the CompNN using C++. (2) When a dataset has a large number of training instances, we used bottleneck features to narrow the training set. Especially in the segmentation problem, we can generate OK results with only 5-10 training reference. (3) Our implementation uses several threads in order to speedup the process. Specifically, each thread is in charge of synthesizing a disjoint set of pixels.

The synthesis of facades using the Facades dataset (400 training samples) takes about 2 hours with 20-30 threads on the CPU. This can be used as a reference for experiments on other datasets.